A Multi-task Learning Framework for Quality Estimation

Sourabh Deoghare[1], Paramveer Choudhary[1], Diptesh Kanojia[1,2],
Tharindu Ranasinghe[3], Pushpak Bhattacharyya[1] and Constantin Orăsan[2]
[1]CFILT, Indian Institie of Technology Bombay, Mumbai, India.
[2]Surrey Institute for People-Centred AI, University of Surrey, United Kingdom.
[3]Aston University, Birmingham, United Kingdom.
{sourabhdeoghare, paramvc, pb}@cse.iitb.ac.in
{d.kanojia, c.orasan}@surrey.ac.uk
{t.ranasinghe}@aston.ac.uk

Abstract

Quality Estimation (QE) is the task of evaluating machine translation output in the absence of reference translation. Conventional approaches to QE involve training separate models at different levels of granularity *viz.,* word-level, sentence-level, and document-level, which sometimes lead to inconsistent predictions for the same input. To overcome this limitation, we focus on jointly training a single model for sentence-level and word-level QE tasks in a multi-task learning framework. Using two multi-task learning-based QE approaches, we show that multi-task learning improves the performance of both tasks. We evaluate these approaches by performing experiments in different settings, *viz.,* single-pair, multi-pair, and zero-shot. We compare the multi-task learning-based approach with baseline QE models trained on single tasks and observe an improvement of up to 4.28% in Pearson's correlation (r) at sentence-level and 8.46% in F1-score at word-level, in the single-pair setting. In the multi-pair setting, we observe improvements of up to 3.04% at sentence-level and 13.74% at word-level; while in the zero-shot setting, we also observe improvements of up to 5.26% and 3.05%, respectively. We make the models proposed in this paper publically available[1].

1 Introduction

Quality Estimation (QE) is a sub-task in the Machine Translation (MT) field. It facilitates the evaluation of MT output without a reference translation by predicting its quality rather than finding its similarity with the reference (Specia et al., 2010). QE is performed at different levels of granularity, *viz.,* word-level QE (Ranasinghe et al., 2021), sentence-level QE (Ranasinghe et al., 2020b), and document-level QE (Ive et al., 2018).

In the sentence-level QE task, current models predict the z-standardized Direct Assessment (DA) score when a source sentence and its translation are provided as inputs. The DA score is a number in the range of 0 to 100, denoting the quality of the translation, obtained from multiple human annotators. These scores are then standardized into z-scores, which are used as labels to train the QE model (Graham et al., 2016).

Unlike the sentence-level QE task, the word-level QE task consists of training a model to predict the 'OK' or 'BAD' tag for each token in a source sentence and its translation. These tags are obtained automatically by comparing the translation with its human post-edits using a token-matching approach. Each source sentence token is tagged as 'OK' if its translation appears in the output and is tagged as 'BAD' otherwise. Similarly, a translation token is assigned an 'OK' tag if it is a correct translation of a source sentence token, and 'BAD' otherwise. Apart from the tokens in the translation, the gaps between the translation tokens are also assigned OK/BAD tags. In case of missing tokens, the gap is tagged as 'BAD', and 'OK' otherwise (Logacheva et al., 2016).

To perform each of these tasks, various deep learning-based approaches are being used (Zerva et al., 2022). While these approaches achieve acceptable performance by focusing on a single task, the learning mechanism ignores information from other QE tasks that might help it do better. By sharing information across related tasks, one can essentially expect the task performance to improve, especially when the tasks are closely related as is the case with the sentence-level and word-level QE. Also, having a separate model for each QE task can cause problems in practical scenarios, like having higher memory and computational requirements. In addition, the different models can produce conflicting information e.g. high DA score, but many errors at word level.

In this paper, we utilize two multi-task learning (MTL)-based (Ruder, 2017) approaches for

[1] https://github.com/cfiltnlp/QE_MTL

Findings of the Association for Computational Linguistics: ACL 2023, pages 9191–9205
July 9-14, 2023 ©2023 Association for Computational Linguistics

word-level and sentence-level QE tasks with the help of a single deep neural network-based architecture. We perform experiments with existing QE datasets (Specia et al., 2020; Zerva et al., 2022) with both MTL approaches to combine word-level and sentence-level QE tasks. We test the following scenarios: **a)** single-pair QE, **b)** multi-pair QE, and **c)** zero-shot QE. The code and models are made available to the community via GitHub.

To the best of our knowledge, we introduce a novel application of the Nash-MTL (Navon et al., 2022) method to both tasks in Quality Estimation. Our **contributions** are:

1. showing that jointly training a single model using MTL for sentence and word-level QE tasks improves performance on both tasks. In a single-pair setting, we observe an improvement of up to 3.48% in Pearson's correlation (r) at the sentence-level and 7.17% in F1-score at the word-level.

2. showing that the MTL-based QE models are significantly more consistent, on word-level and sentence-level QE tasks for same input, as compared to the single-task learning-based QE models.

We discuss the existing literature in Section 2 and the datasets used in Section 3. The MTL-based QE approach is presented in Section 4. The experimental setup is described in 5. Section 6 discusses the results in detail, including a qualitative analysis of a few sample outputs. We conclude this article in Section 7, where we also propose future research directions in the area.

2 Related Work

During the past decade, there has been tremendous progress in the field of machine translation quality estimation, primarily as a result of the shared tasks organized annually by the Conferences on Machine Translation (WMT), since 2012. These shared tasks have produced benchmark datasets on various aspects of quality estimation, including word-level and sentence-level QE. Furthermore, these datasets have led to the development and evaluation of many open-source QE systems like QuEst (Specia et al., 2013), QuEst++ (Specia et al., 2015), deepQuest (Ive et al., 2018), and OpenKiwi (Kepler et al., 2019). Before the neural network era, most of the quality estimation systems like QuEst (Specia et al., 2013), and QuEst++ (Specia et al., 2015) were heavily dependent on linguistic processing and feature engineering to train traditional machine-learning algorithms like support vector regression and randomized decision trees (Specia et al., 2013).

In recent years, neural-based QE systems such as deepQuest (Ive et al., 2018), and OpenKiwi (Kepler et al., 2019) have consistently topped the leaderboards in WMT quality estimation shared tasks (Kepler et al., 2019). These architectures revolve around an encoder-decoder Recurrent Neural Network (RNN) (referred to as the 'predictor'), stacked with a bidirectional RNN (the 'estimator') that produces quality estimates. One of the disadvantages of this architecture is they require extensive predictor pre-training, which means it depends on large parallel data and is computationally intensive (Ive et al., 2018). This limitation was addressed by TransQuest (Ranasinghe et al., 2020b), which won the WMT 2020 shared task on sentence-level DA. TransQuest eliminated the requirement for predictor by using cross-lingual embeddings (Ranasinghe et al., 2020b). The authors fine-tuned an XLM-Roberta model on a sentence-level DA task and showed that a simple architecture could produce state-of-the-art results. Later the TransQuest framework was extended to the word-level QE task (Ranasinghe et al., 2021).

A significant limitation of TransQuest is that it trains separate models for word-level and sentence-level QE tasks. While this approach has produced state-of-the-art results, managing two models requires more computing resources. Furthermore, since the two models are not interconnected, they can provide conflicting predictions for the same translation. To overcome these limitations, we propose a multi-task learning approach to QE.

Multitask architectures have been employed in several problem domains, such as those in computer vision (Girshick, 2015; Zhao et al., 2018) and natural language processing (NLP). In NLP, tasks such as text classification (Liu et al., 2017), natural language generation (Liu et al., 2019), part-of-speech tagging and named entity recognition (Collobert and Weston, 2008) have benefited from MTL. In QE too, Kim et al. (2019) has developed an MTL architecture using a bilingual BERT model. However, the model does not provide results similar to or better than state-of-the-art QE frameworks such as TransQuest (Ranasinghe et al., 2021). Some of the recent WMT QE shared task submissions also use MTL to develop QE systems (Specia et al., 2020, 2021; Zerva et al., 2022). As all these submissions are not evaluated under the same experimental settings and use different techniques along with MTL, the improvements due to MTL alone are difficult to assess. In this paper, we introduce a novel MTL approach for QE that outperforms TransQuest in both word-level and sentence-level QE tasks, in various experimental settings.

3 Datasets: WMT 2022

We use data provided in the WMT21 (Specia et al., 2021), and WMT22 (Zerva et al., 2022) Quality Estimation Shared tasks for our experiments. We choose language pairs for which word-level and sentence-level annotations are available for the same source-translation pairs. The data consists of three low-resource language pairs: English-Marathi (En-Mr), Nepali-English (Ne-En), Sinhalese-English (Si-En); three medium-resource language pairs: Estonian-English (Et-En), Romanian-English (Ro-En), Russian-English (Ru-En); and one high-resource language pair: English-German (En-De). For the English-Marathi language pair, the data consists of 20K training instances and 1K instances each for validation and testing[2]. The training set consists of 7K instances for all other language pairs, and validation and test sets consist of 1K samples each.

Each sample in the word-level QE data for any language pair except English-Marathi consists of a source sentence, its translation, and a sequence of tags for tokens and gaps. For the English-Marathi pair, the WMT22 dataset does not contain tags for gaps in tokens. Therefore, we used the QE corpus builder[3] to obtain annotations for translations using their post-edited versions.

4 Approach

In this section, we briefly discuss the TransQuest framework, explain the architecture of our neural network, and then discuss the MTL approaches we used for the experimentation, along with the mathematical modeling.

4.1 TransQuest Framework

We use the MonoTransQuest (for sentence-level QE model) (Ranasinghe et al., 2020b) and MicroTransQuest (for word-level QE model) (Ranasinghe et al., 2021) architectures to perform the single-task-based QE experiments. The MonoTransQuest architecture (1) uses a single XLM-R (Conneau et al., 2020) transformer model. The input of this model is a concatenation of the original sentence and its translation. Both these sequences are separated by a special [SEP] token. The inputs are passed to an embedding layer to obtain embeddings for each token. The Direct Assessment (DA) scores are produced by passing the output of the [CLS] token through a softmax layer.

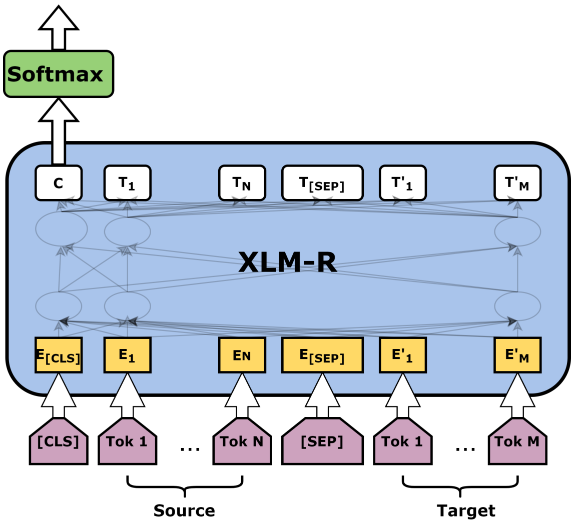


Figure 1: Architecture of the MonoTransQuest sentence-level QE Model.

Similarly, the MicroTransquest architecture presented in figure 2 also uses the XLM-R transformer. The input to this model is a concatenation of the original sentence and its translation, separated by the [SEP] token. Additionally, the [GAP] tokens are added between the translation tokens. Finally, an output of each token is passed through a softmax layer to obtain the OK or BAD tag for each token.

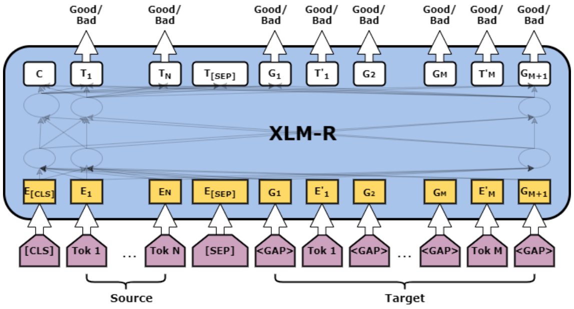


Figure 2: Architecture of the MicroTransQuest word-level QE Model.

[2] https://github.com/surrey-nlp/IndicQE-PE

[3] https://github.com/deep-spin/qe-corpus-builder

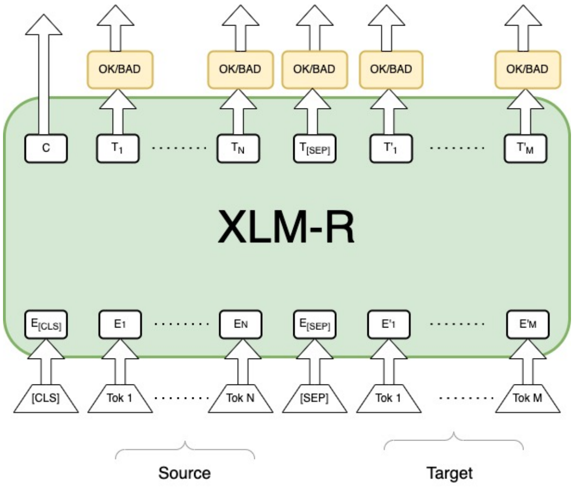


Figure 3: Architecture of the MTL QE model.

4.2 Model Architecture

Considering the success that transformers have demonstrated in translation quality estimation (Ranasinghe et al., 2020a; Wang et al., 2021), we chose to employ the transformer as a base model for our MTL approach. Our approach learns two tasks jointly: sentence-level and word-level quality estimation.

Figure 3 depicts the model's architecture used in our approach. The implemented architecture shares hidden layers between both sentence-level and word-level QE tasks. The shared portion includes the XLM-Roberta (Conneau et al., 2020) model that learns shared representations (and extracts information) across tasks by minimizing a combined/compound loss function. The task-specific heads receive input from the last hidden layer of the transformer language model and predict the output for each task (details provided in the next two sections).

Sentence-level Quality Estimation Head By utilizing the hidden representation of the classification token (CLS) within the transformer model, we predict the DA scores by applying a linear transformation:

$$\hat{\mathbf{y}}_{da} = \mathbf{W}_{[CLS]} \cdot \mathbf{h}_{[CLS]} + \mathbf{b}_{[CLS]} \qquad (1)$$

where $\cdot$ denotes matrix multiplication, $\mathbf{W}_{[CLS]} \in \mathcal{R}^{D\times 1}$, $\mathbf{b}_{[CLS]} \in \mathcal{R}^{1\times 1}$, and D is the dimension of input layer $\mathbf{h}$ (top-most layer of the transformer).

Word-level Quality Estimation Head We predict the word-level labels (OK/BAD) by applying a linear transformation (also followed by the softmax) over every input token from the last hidden layer of the model:

$$\hat{\mathbf{y}}_{word} = \sigma(\mathbf{W}_{token} \cdot \mathbf{h}_t + \mathbf{b}_{token}) \qquad (2)$$

where t marks which token the model is to label within a T-length window/token sequence, $\mathbf{W}_{token} \in \mathcal{R}^{D\times 2}$, and $\mathbf{b}_{token} \in \mathcal{R}^{1\times 2}$. This part is similar to the MicroTransQuest architecture in Ranasinghe et al. (2021).

4.3 Multi-Task Learning

We use two MTL approaches to train the QE models. In the first approach, task-specific losses are combined into a single loss by summing them. The second approach considers the gradient conflicts and follows a heuristic-based approach to decide the update direction.

Linear Scalarization (LS) We train the system by minimizing the Mean Squared Error (MSE) for the sentence-level QE task and cross-entropy loss for the word-level QE task as defined in Equation 3 and Equation 4, where $\mathbf{y}_{da}$ and $\mathbf{y}_{word}$ represent ground true labels. These particular losses are:

$$\mathcal{L}_{da} = MSE\Big(\mathbf{y}_{da}, \hat{\mathbf{y}}_{da}\Big) \qquad (3)$$

$$\mathcal{L}_{word} = -\sum_{i=1}^{2} \Big(\mathbf{y}_{word} \odot \log(\hat{\mathbf{y}}_{word})\Big)[i] \qquad (4)$$

where $\mathbf{v}[i]$ retrieves the ith item in a vector $\mathbf{v}$ and $\odot$ indicates element-wise multiplication. For combining the above two losses into one objective, α and β parameters are used to balance the importance of the tasks. n this study, we assign equal importance to each task in our experiments, therefore we set $\alpha = \beta = 1$ in this study. The final loss is shown in Equation 5.

$$\mathcal{L}_{MultiTransQuest} = \frac{\alpha\mathcal{L}_{da} + \beta\mathcal{L}_{word}}{\alpha + \beta} \qquad (5)$$

We set up two baselines – single-task learning-based sentence-level QE and word-level QE models. The sentence-level QE model takes a source sentence and its translation as input and predicts the DA score. We use the MonoTransQuest implementation in Ranasinghe et al. (2020b) for this sentence-level QE model. The word-level QE model predicts whether each token (word) is OK or BAD using a softmax classifier as well. We use the MicroTransQust implementation in Ranasinghe et al. (2021) as the word-level QE model.

Nash Multi-Task Learning (Nash-MTL) Joint training of a single model using multi-task learning is known to lower computation costs. However, due to potential conflicts between the gradients of different tasks, the joint training typically results in the jointly trained model performing worse than its equivalent single-task counterparts. Combining per-task gradients into a combined update direction using a specific heuristic is a popular technique for solving this problem. In this approach, the tasks negotiate for a joint direction of parameter update.

Algorithm 1 Nash_MTL

```
Input: $\theta_0$ - initial parameter vector, $\{l_i\}_{i=1}^{K}$ - differentiable loss functions, $\eta$ - learning rate
Output: $\theta^T$
for $t = 1,...,T$ do
    Compute task gradients $g_i^t = \nabla_{\theta(t-1)} l_i$
    Set $G^{(t)}$ the matrix with columns $g_i^{(t)}$
    Solve for $\alpha$ : $(G^t)^T(G^t)\alpha = 1/\alpha$ to obtain $\alpha^t$
    Update the parameters $\theta^{(t)} = \theta^{(t)} - \eta G^{(t)}\alpha^{(t)}$
end for
return $\theta^T$
```

For the MTL problem with parameters θ, the method assumes a sphere B_ϵ, with a center at zero and a radius ϵ. The update vectors $\Delta\theta$ are searched inside this sphere. The problem is framed as a bargaining problem by considering the centre as the point of disagreement and the B_ϵ as an agreement set. For every player, the utility function is $u_i(\Delta\theta) = g_i^T \Delta\theta$ where g_i denotes the gradient vector at θ of the loss of task i. Additional details, theoretical proof and empirical results on various tasks can be followed from Navon et al. (2022), who proposed this gradient combination.

5 Experimental Setup

This section describes the different experiments we perform and the metrics we use to evaluate our approach. We also discuss the training details and mention the computational resources used for training the models.

Experiments We perform our experiments under three settings: single-pair, multi-pair, and zero-shot. For each setting, we train one sentence-level, one word-level, and two MTL-based QE models. The first two models are the Single-Task Learning (STL)-based QE models (STL QE), and we use their performance as baselines. The TransQuest framework (Ranasinghe et al., 2020b) contains the MonoTransQuest model for the sentence-level QE task and the MicroTransQuest model (Ranasinghe et al., 2021) for word-level QE task which helped us reproduce baseline results over all the language pairs investigated for this paper. The next two models are the MTL-based QE models (MTL QE) trained using two different MTL approaches explained in Section 4. For training LS models, we use the Framework for Adapting Representation Models (FARM)[4], while for training Nash-MTL models, we used implementation[5] shared by the authors. All the experiments use all seven language pairs introduced in Section 3.

In the single-pair setting, we only use the data of one particular language pair for training and evaluation. However, in the multi-pair setting, we combine training data of all the language pairs and evaluate the model using test sets of all language pairs. For the transfer-learning experiments (zero-shot setting), we combine training data of all language pairs except the language pair on which we evaluate the model.

Evaluation We use the Pearson Correlation (r) between the predictions and gold-standard annotations for evaluating the sentence-level QE as it is a regression task. Similarly, for the word-level QE, which is treated as a token-level classification task, we consider the F1-score as an evaluation metric. We perform a statistical significance test considering primary metrics using William's significance test (Graham, 2015).

Training Details To maintain uniformity across all the languages, we used an identical set of settings for all the language pairings examined in this work. For the STL and LS-MTL models, we use a batch size of 16. We start with a learning rate of $2e - 5$ and use 5% of training data for warm-up. We use early stopping and patience over the 10 steps. The Nash-MTL models are trained using the configuration outlined in (Navon et al., 2022). Considering the availability of computational resources, the STL QE models are trained using the NVIDIA RTX A5000 GPUs, and the MTL QE models using the NVIDIA DGX A100 GPUs. Additional training details are provided in **Appendix A**.

[4] https://github.com/deepset-ai/FARM

[5] https://github.com/AvivNavon/nash-mtl

LP	Word-Level					Sentence-Level				
	STL	LS-MTL	+/- %	Nash-MTL	+/- %	STL	LS-MTL	+/- %	Nash-MTL	+/- %
En-Mr	0.3930	0.4194	2.64%	**0.4662**	7.32%	0.5215	0.5563	3.48%	**0.5608**	3.93%
Ne-En	0.4852	0.5383	5.31%	**0.5435**	5.83%	0.7702	0.7921	2.19%	**0.8005**	3.03%
Si-En	0.6216	0.6556	3.40%	**0.6946**	7.30%	0.6402	0.6533	1.31%	**0.6791**	3.89%
Et-En	0.4254	0.4971	7.17%	**0.5100**	8.46%	0.7646	0.7905	2.59%	**0.7943**	2.97%
Ro-En	0.4446	0.4910	4.64%	**0.5273**	8.27%	0.8952	**0.8985***	0.33%	0.8960*	0.08%
Ru-En	0.3928	0.4208	2.80%	**0.4394**	4.66%	0.7864	0.7994	1.30%	**0.8000**	1.36%
En-De	0.3996	0.4245	2.49%	**0.4467**	4.71%	0.4005	0.4310	3.05%	**0.4433**	4.28%

Table 1: Results obtained for **word-level (F1-scores) and sentence-level (Pearson (r)) QE tasks in the single-pair** setting. **STL**: results from the models trained using TransQuest. **LS-MTL** and **Nash-MTL**: results obtained using the Linear Scalarization MTL approach, and the Nash-MTL-based models, respectively. The first three rows show results for the low-resource language pairs, the next three for mid-resource, and the last for a high-resource language pair. [(*) indicates the improvement is not significant with respect to the baseline score.]

LP	Word-Level $(F1)$					Sentence-Level (r)				
	STL	LS-MTL	+/- %	Nash-MTL	+/- %	STL	LS-MTL	+/- %	Nash-MTL	+/- %
En-Mr	0.4013	0.4349	3.36%	**0.4815**	8.02%	**0.6711**	0.6514*	-1.97%	0.6704*	-0.07%
Ne-En	0.4902	0.5406	5.04%	**0.5560**	6.58%	0.7892	**0.8012**	1.20%	0.8001	1.09%
Si-En	0.5629	0.6392	7.63%	**0.7003**	13.74%	0.6653	0.6837	1.84%	**0.6957**	3.04%
Et-En	0.4348	0.4998	6.50%	**0.5082**	7.34%	0.7945	**0.7970***	0.25%	0.7963*	0.18%
Ro-En	0.4472	0.4925	4.53%	**0.5285**	8.13%	**0.8917**	0.8883*	-0.34%	0.8895*	-0.22%
Ru-En	0.3965	**0.4241**	2.76%	0.4211	2.46%	0.7597	0.7751	1.54%	**0.7772**	1.75%
En-De	0.3972	0.4253	2.81%	**0.4499**	5.27%	**0.4373**	0.4308*	-0.65%	0.4298*	-0.75%

Table 2: Results obtained for **word-level and sentence-level QE tasks in the multi-pair** setting. [* indicates the improvement is not significant with respect to the baseline score.]

6 Results and Discussion

Results of the single-pair, multi-pair, and zero-shot settings are presented in this section. The tables referred to in this section report performance of the STL, LS-MTL, and Nash-MTL QE models using the Pearson correlation (r) and F1-score for sentence-level and word-level QE, respectively.

We could not conduct a direct performance comparison between our QE models and winning entries of the recent WMT QE shared tasks due to the following reasons: **(1)** Nature of the word-level QE task, and its evaluation methodology have changed over the years. Until last year, gaps between translation tokens were a part of the data, and the 'OK' or 'BAD' tags were predicted for them as well. But the WMT22 shared task did not consider these gaps; and **(2)** Not all the language pairs investigated in this paper have been a part of WMT QE tasks in the same year. Therefore, we establish a standard baseline using the Transformers-based framework, TransQuest, and show improvements.

We also compare Pearson correlation coefficients obtained by STL and MTL QE models to assess whether MTL QE model predictions on both tasks for the same inputs are consistent (Table 4). Furthermore, we perform a qualitative analysis of the output for En-Mr, Ro-En, and Si-En language pairs, and show some examples in Table 5. We discuss the analysis in detail in subsection 6.4.

6.1 Single-Pair Setting

The results for the first experimental setting are presented in Table 1. The MTL QE approaches provide significant performance improvements for all language pairs in the sentence and word-level QE tasks over the respective STL QE models. In the word-level QE task, the Nash-MTL QE models outperform the STL and LS-MTL models for all language pairs. Our approach achieves the highest improvement of 8.46% in terms of macro F1-score for the Et-En language pair. While for the En-De, we observe the least improvement from the LS-MTL QE model (2.49%). The average improvement in the F1-score from Nash-MTL model and LS-MTL model is 6.29% and 4.06%, respectively.

LP	Word-Level					Sentence-Level				
	STL	LS-MTL	+/- %	Nash-MTL	+/- %	STL	LS-MTL	+/- %	Nash-MTL	+/- %
En-Mr	0.3800	0.3692*	-1.08%	**0.3833**	0.33%	0.4552*	0.3869	-6.83%	**0.4674**	1.22%
Ne-En	0.4175	0.4472	2.97%	**0.4480**	3.05%	0.7548	**0.7601**	0.53%	0.7560	0.12%
Si-En	0.4239	0.4250*	0.11%	**0.4407**	1.68%	0.6416	0.6434*	0.18%	**0.6447***	0.31%
Et-En	0.4049	0.4206	1.57%	**0.4291**	2.42%	0.5192	0.5583	3.91%	**0.5598**	4.06%
Ro-En	0.4179	0.4349	1.70%	**0.4420**	2.41%	0.5962	0.6104	1.42%	**0.6300**	3.38%
Ru-En	0.3737	0.3761*	0.24%	**0.3834**	0.97%	0.5286	0.5605	3.19%	**0.5812**	5.26%
En-De	0.3750	0.3763*	0.13%	**0.3768***	0.18%	0.3217	0.3227*	0.10%	**0.3305**	0.88%

Table 3: Results obtained for **word-level and sentence-level QE tasks in the zero-shot** setting. [* indicates the improvement is not significant with respect to the baseline score.]

For the sentence-level QE task, Pearson correlation (r) between the QE system prediction scores and true labels is used as an evaluation metric. For this task, the MTL QE models, again, outperform the STL QE models for all language pairs. Here, the En-De Nash-MTL QE model obtains the most significant performance improvement of 4.28% over the corresponding STL QE model. A minor performance improvement of 0.33% is observed for the Ro-En language pair using the LS-MTL QE model. The average improvement in Pearson's correlation (r) from the Nash-MTL model and the LS-MTL model is 2.75% and 2.10%, respectively.

Except for the Ro-En Nash-MTL QE model's performance in the sentence-level QE task, we see the Nash-MTL QE models amass the most improvements over the STL and LS-MTL QE models for all language pairs in both tasks. It shows that the bargaining between the gradient update directions for sentence-level and word-level QE tasks that the Nash-MTL method arranges results in effective learning. The results of both tasks also show that we get more improvements for low-resource and mid-resource language pairs than for the high-resource language pair.

We additionally report the results obtained by the WMT QE shared task winning systems in **Appendix C**. The WMT figures are not directly comparable to our results. The WMT figures are higher than ours but that is really not the point. Our aim is to show that multitask learning is more effective than single-task learning. Any QE technique can seriously be considered adopting MTL in preference to the STL. Of course, if the STL figures are already high then the improvement may not be significant which we also have observed.

6.2 Multi-Pair Setting

Table 2 tabulates the results for the multi-pair setting. The multi-pair setting can benefit the word-level QE task due to vocabulary overlap and the sentence-level QE tasks due to syntactical similarities between the language pairs.

In this setting, MTL improves performance for all language pairs in the word-level QE task. Using the LS-MTL QE model, the highest F1-score improvement of 7.63% is observed for the Si-En language pair, while with the Nash-MTL QE model, the best improvement is of 13.74%. The least improvement with the LS-MTL QE model is observed for the Ru-En pair 2.76%, while for the Nash-MTL-based QE model, it is of 2.46% for the Ru-En pair.

Though the improvements observed in the word-level QE task in this setting when using MTL QE approaches are even higher compared to the single-pair setting, we see an opposite trend in the sentence-level QE task results. At the sentence level, we observe a slight degradation in the results of the En-Mr, En-De, and Ro-En MTL QE models. We observe the most improvement of the 3.04% in Pearson Correlation over the STL QE model by the Nash-MTL QE model. For the Ro-En pair, both QE models fail to bring improvements over the STL QE model. For Ne-En and Et-En pairs, the LS-MTL QE model outperforms the Nash-MTL QE model. In this setting, the Nash-MTL technique provides similar results to the LS-MTL technique. Also, we observe that the Nash-MTL QE approach benefits the most to the low-resource language pairs. We also see higher improvements for the mid-resource language pairs than the high-resource language pair.

LP	Pearson Correlation (r)			Spearman Correlation (ρ)		
	STL	Nash-MTL	+/-	STL	Nash-MTL	+/-
En-Mr	-0.2309	-0.3645	**13.36%**	-0.1656	-0.2963	**13.07%**
Ne-En	-0.6263	-0.6604	3.41%	-0.6124	-0.6442	3.18%
Si-En	-0.5522	-0.5881	3.59%	-0.5380	-0.5510	1.30%
Et-En	-0.7202	-0.7539	3.37%	-0.7541	-0.768	1.39%
Ro-En	-0.7765	-0.7794	0.29%	-0.7380	-0.7534	1.54%
Ru-En	-0.6930	-0.7187	2.57%	-0.6364	-0.6805	4.41%
En-De	-0.4820	-0.5482	6.62%	-0.4524	-0.5099	5.75%

Table 4: Pearson (r) and Spearman (ρ) correlations between sentence-level and word-level QE predictions using STL and Nash-MTL QE models. The sentence-level QE prediction is the z-standardized Direct Assessment (DA) score, and the word-level QE prediction is the bad tag count normalized by sentence length.

6.3 Zero-shot Setting

Table 3 shows the results for the zero-shot setting. The MTL QE models achieve better performance for both tasks over their STL-based counterparts for all the language pairs, except for the En-Mr language pair in the sentence-level QE task. Surprisingly, for the Ne-En pair, the LS-MTL model outperforms the Nash-MTL QE model in the sentence-level QE task by a small margin (0.0053). While for all other language pairs, the Nash-MTL QE models outperform the respective LS-MTL QE models. Similar to the trend in the previous two settings, the MTL QE approaches bring more benefits to the low-resource and mid-resource language pairs than the high-resource language pair.

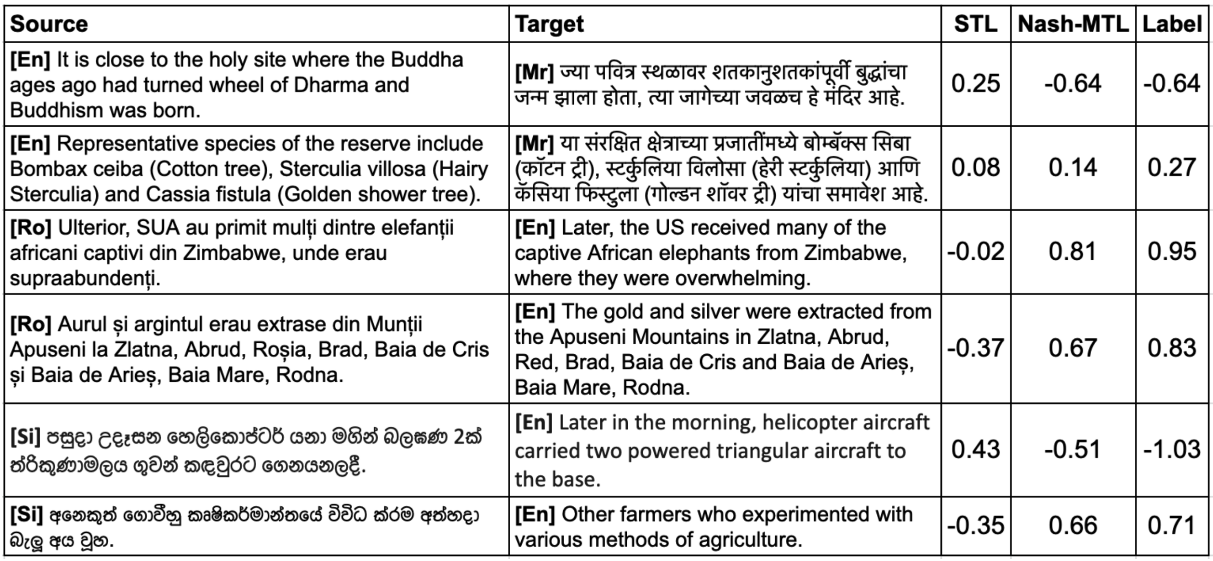
In **Appendix B**, for each low-resource language pair, we include a table showing the comparison of STL, LS-MTL, and Nash-MTL QE models. These tables show that *the multi-pair setting helps the low-resource scenario*.

6.4 Discussion

Consistent Predictions Improvements shown by the MTL QE models in varied experimental settings on both tasks show that the tasks complement each other. We further assess the potential of the MTL QE models in predicting consistent outputs for both tasks over the same inputs. We do so by computing a correlation between the predicted DA scores and the percentage of tokens in a sentence for which the 'BAD' tag was predicted. Therefore, a *stronger negative correlation denotes better consistency*. Table 4 shows Pearson and Spearman correlations between sentence-level and word-level QE predictions on the test sets, in a single pair setting. For all the language pairs, Nash-MTL QE models show a stronger correlation than the STL QE models. We also perform a qualitative analysis of the STL and MTL QE models for the En-Mr, Ro-En, and Si-En language pairs.

Qualitative Analysis The first English-Marathi example is shown in Table 5. It contains a poor translation of the source sentence meaning, "The temple is close to the holy place where ages ago the Buddha was born." The STL word-level QE and MTL QE models predict the same output assigning correct tags to tokens, yet we observe a significant difference in the sentence-level scores predicted by the models. The STL sentence-level QE model outputs a high score of 0.25, while the score given by the MTL QE model is -0.64. It supports the observation that the *MTL QE model outputs are more consistent*.

Unlike the STL sentence-level QE models, the MTL QE models predict more justified quality scores when translations have only minor mistakes. The translation in the first Ro-En example in Table 5 is a high-quality translation. In this translation, the word "overwhelming" could have been replaced with a better lexical item. The STL QE model harshly penalizes the translation by predicting the z-score at -0.0164, while the MTL model predicts a more justifiable score (0.8149). Similar behaviour is reflected in the second Si-En example as well (last row). Even though the translation reflects the meaning of the source sentence adequately and is also fluent, the STL QE model predicts a low score of -0.35, while the MTL QE



Source	Target	STL	Nash-MTL	Label
[En] It is close to the holy site where the Buddha ages ago had turned wheel of Dharma and Buddhism was born.	**[Mr]** ज्या पवित्र स्थळावर शतकानुशतकांपूर्वी बुद्धांचा जन्म झाला होता, त्या जागेच्या जवळच हे मंदिर आहे.	0.25	-0.64	-0.64
[En] Representative species of the reserve include Bombax ceiba (Cotton tree), Sterculia villosa (Hairy Sterculia) and Cassia fistula (Golden shower tree).	**[Mr]** या संरक्षित क्षेत्राच्या प्रजातींमध्ये बोम्बॅक्स सिबा (कॉटन ट्री), स्टर्कुलिया विलोसा (हेरी स्टर्कुलिया) आणि कॅसिया फिस्टुला (गोल्डन शॉवर ट्री) यांचा समावेश आहे.	0.08	0.14	0.27
[Ro] Ulterior, SUA au primit mulți dintre elefanții africani captivi din Zimbabwe, unde erau supraabundenți.	**[En]** Later, the US received many of the captive African elephants from Zimbabwe, where they were overwhelming.	-0.02	0.81	0.95
[Ro] Aurul și argintul erau extrase din Munții Apuseni la Zlatna, Abrud, Roșia, Brad, Baia de Cris și Baia de Arieș, Baia Mare, Rodna.	**[En]** The gold and silver were extracted from the Apuseni Mountains in Zlatna, Abrud, Red, Brad, Baia de Cris and Baia de Arieș, Baia Mare, Rodna.	-0.37	0.67	0.83
[Si] පසුදා උදෑසන හෙලිකොප්ටර් යානා මගින් බලඝණ 2ක් ත්‍රිකුණාමලය ගුවන් කඳවුරට ගෙනයනලදී.	**[En]** Later in the morning, helicopter aircraft carried two powered triangular aircraft to the base.	0.43	-0.51	-1.03
[Si] අනෙකුත් ගොවීහු කෘෂිකර්මාන්තයේ විවිධ ක්‍රම අත්හදා බැලූ අය වූහ.	**[En]** Other farmers who experimented with various methods of agriculture.	-0.35	0.66	0.71

Table 5: Source-translation pairs along with z-standardized DA scores by STL, Nash-MTL QE models, and the ground truth labels.

model rates the translation appropriately by predicting 0.66 as score.

We also observed that the MTL QE models have an edge when rating translations with many named entities. This can be seen through the second English-Marathi (Row 3), second Romanian-English (Row 5), and first Sinhala-English (Row 6) examples in Table 5. The translations are of high quality in both examples, and the MTL QE models rate them more appropriately than the STL QE models.

7 Conclusion and Future Work

In this paper, we showed that jointly training a single, pre-trained cross-lingual transformer over the sentence-level and word-level QE tasks improves performance on both tasks. We evaluated our approach in three different settings: single-pair, multi-pair, and zero-shot. The results on both the QE tasks show that the MTL-based models outperform their STL-based counterparts for multiple language pairs in the single-pair setting. Given the performance in the zero-shot setting, we see promising transfer-learning capabilities in our approach. Consistent scores across both QE tasks for the same inputs demonstrate the effectiveness of the MTL method to QE. We release our MTL-based QE models and our code under the CC-BY-SA 4.0 license publicly for further research.

In future, we wish to extend this work and evaluate the MTL-based QE models in a few-shot setting to assess the effectiveness of transfer learning. Further, we would like to explore the usage of word-level QE and sentence-level QE to assist in the task of automatic post-editing for MT. We also wish to explore the use of language-relatedness for building multi-pair MTL-based QE models.

Limitations

The experimental results suggest the possibility of our MTL-based QE approach being biased towards the word-level QE task, as the jointly trained QE models show better performance improvements for the word-level QE task as compared to the sentence-level QE task. Further, we also observe that our approach does not work well for language pairs with English as a source language (En-De and En-Mr). The qualitative analysis of the English-Marathi MTL-based QE model shows that the model performs poorly when inputs are in the passive voice. Our multi-pair setting experiments use all seven language pairs. We do not consider properties like the similarity between the languages, translation directions, *etc.*, to group the language pairs. So it may be possible to achieve comparable performance using a subset of languages. We choose the Nash-MTL approach for MTL-based experiments because it has been compared with around ten other MTL techniques and it has been shown that the Nash-MTL approach outperforms them on different combinations of the tasks. In the current work, we have not experimentally analyzed how the Nash-MTL approach gives better improvements than the LS-MTL approach.

Ethics Statement

Our MTL architectures are trained on multiple publicly available datasets referenced in this paper. These datasets have been previously collected and annotated, and no new data collection has been carried out as part of this work. Furthermore, these are standard benchmarks that have been released in recent WMT shared tasks. No user information was present in the datasets protecting users' privacy and identity. We understand that every dataset is subject to intrinsic bias and that computational models will inevitably learn biased information from any dataset. That said, we also believe that our MTL models will help diminish biases in QE as they provide an explainable aspect to the predictions through token-level labels.

References

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ross Girshick. 2015. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448.

Yvette Graham. 2015. Improving evaluation of machine translation quality estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1804–1813, Beijing, China. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Meghan Dowling, Maria Eskevich, Teresa Lynn, and Lamia Tounsi. 2016. Is all that glitters in machine translation quality estimation really gold? In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3124–3134, Osaka, Japan. The COLING 2016 Organizing Committee.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. deepQuest: A framework for neural-based quality estimation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3146–3157, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Hyun Kim, Joon-Ho Lim, Hyun-Ki Kim, and Seung-Hoon Na. 2019. QE BERT: Bilingual BERT using multi-task learning for neural quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 85–89, Florence, Italy. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Varvara Logacheva, Chris Hokamp, and Lucia Specia. 2016. MARMOT: A toolkit for translation quality estimation at the word level. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3671–3674, Portorož, Slovenia. European Language Resources Association (ELRA).

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. 2022. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, pages 16428–16446. PMLR.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020a. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020b. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2021. An exploratory analysis of multilingual word-level quality estimation with cross-lingual transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 434–440, Online. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with QuEst++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine translation*, 24(1):39–50.

Lucia Specia, Kashif Shah, Jose G.C. de Souza, and Trevor Cohn. 2013. QuEst - a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sofia, Bulgaria. Association for Computational Linguistics.

Jiayi Wang, Ke Wang, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021. QEMind: Alibaba's submission to the WMT21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 948–954, Online. Association for Computational Linguistics.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin OrÄƒsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the wmt 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation*, pages 69–99, Abu Dhabi. Association for Computational Linguistics.

Xiangyun Zhao, Haoxiang Li, Xiaohui Shen, Xiaodan Liang, and Ying Wu. 2018. A modulation module for multi-task learning with applications in image retrieval. In *Computer Vision – ECCV 2018*, pages 415–432, Cham. Springer International Publishing.

A Additional Training Details

The number of parameters for our STL QE models trained using the TransQuest framework is 125M since we use the XLM-R base model variant for all experiments. This language model has 12 heads, with an embedding dimension of 768. The number of parameters in our MTL QE model is also approximately 125M.

Our total computation time for the STL models was approximately 60 hours, whereas the computation time for all experiments under LS-MTL was approximately 22.5 hours. However, our best-performing approach, *i.e.,* Nash-MTL, took approximately 41.25 hours.

Model	**Setting**	$F1$	r
STL	**Single-Pair**	0.3930	0.5215
	Multi-Pair	0.4013	0.6711
	Zero-Shot	0.3800	0.4552
LS-MTL	**Single-Pair**	0.4194	0.5563
	Multi-Pair	0.4349	0.6514
	Zero-Shot	0.3692	0.3869
Nash-MTL	**Single-Pair**	0.4662	0.5608
	Multi-Pair	**0.4815**	**0.6704**
	Zero-Shot	0.3833	0.4674

Table 6: Results obtained for the **En-Mr** Language pair.

Model	**Setting**	**F1**	**r**
STL	**Single-Pair**	0.4852	0.7702
	Multi-Pair	0.4902	0.7892
	Zero-Shot	0.4175	0.7548
LS-MTL	**Single-Pair**	0.5383	0.7921
	Multi-Pair	0.5406	**0.8012**
	Zero-Shot	0.4472	0.7601
Nash-MTL	**Single-Pair**	0.5435	0.8005
	Multi-Pair	**0.5560**	0.8001
	Zero-Shot	0.4480	0.7560

Table 7: Results obtained for the **Ne-En** Language pair.

B Low-resource Setting Results

Here, we try to compare the performance of our proposed approaches on low-resource language pairs, in all three settings and for both tasks, in a concise manner. Table 6, Table 7, and Table 8 show that the Nash-MTL-based QE approach in the multi-pair setting outperforms the single-pair settings for all the low-resources languages. Table 6 shows this comparison in terms of F1 for word-level QE and Pearson's (r) for the En-Mr language pair. Table 7, and 8 show the same results for Ne-En and Si-En, respectively.

Model	**Setting**	$F1$	r
STL	**Single-Pair**	0.6216	0.6402
	Multi-Pair	0.5629	0.6653
	Zero-Shot	0.4239	0.6416
LS-MTL	**Single-Pair**	0.6556	0.6533
	Multi-Pair	0.6392	0.6837
	Zero-Shot	0.4250	0.6434
Nash-MTL	**Single-Pair**	0.6946	0.6791
	Multi-Pair	**0.7003**	**0.6957**
	Zero-Shot	0.4407	0.6447

Table 8: Comparison of all models under all three settings for Si-En Language pair.

C Additional Single Pair Setting Results

We additionally report the results of winning submissions to the WMT21 and WMT22 QE shared tasks for the single-pair setting. Table 9 tabulates the results. Results obtained by the winning systems of WMT21 QE shared tasks are reported for all language pairs except English-Marathi. For the English-Marathi pair, we report the result achieved by the WMT22 shared task-winning systems. We report the F1-multi results for the word-level QE task and Pearson's correlation (r) for the sentence-level QE shared task.

LP	**Word-level**						**Sentence-level**					
	STL	**LS-MTL**	**+/- %**	**Nash-MTL**	**+/- %**	**WMT**	**STL**	**LS-MTL**	**+/- %**	**Nash-MTL**	**+/- %**	**WMT**
En-Mr	0.3930	0.4194	2.64%	0.4662	7.32%	0.5827	0.5215	0.5563	3.48%	0.5608	3.93%	0.604
Ne-En	0.4852	0.5383	5.31%	0.5435	5.83%	0.5693	0.7702	0.7921	2.19%	0.8005	3.03%	0.867
Si-En	0.6216	0.6556	3.40%	0.6946	7.30%	0.7140	0.6402	0.6533	1.31%	0.6791	3.89%	0.605
Et-En	0.4254	0.4971	7.17%	0.5100	8.46%	0.5140	0.7646	0.7905	2.59%	0.7943	2.97%	0.812
Ro-En	0.4446	0.4910	4.64%	0.5273	8.27%	0.5777	0.8952	0.8985	0.33%	0.8960	0.08%	0.908
Ru-En	0.3928	0.4208	2.80%	0.4394	4.66%	0.4480	0.7864	0.7994	1.30%	0.8000	1.36%	0.806
En-De	0.3996	0.4245	2.49%	0.4467	4.71%	0.4267	0.4005	0.4310	3.05%	0.4433	4.28%	0.584

Table 9: Results obtained for word-level (F1-scores) and sentence-level (Pearson (r)) QE tasks in the single-pair setting. STL: results from the models trained using TransQuest. LS-MTL and Nash-MTL: results obtained using the Linear Scalarization MTL approach, and the Nash-MTL-based models, respectively. WMT: results obtained by the winning submission of the WMT21/WMT22 shared tasks. The first three rows show results for the low-resource language pairs, the next three for mid-resource, and the last for a high-resource language pair. *Please note that the WMT results are not directly comparable with the LS-MTL or the Nash-MTL results.*

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
It is an unnumbered section on page 8/9.

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction (Section 1)

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We create computational models for the task of Quality Estimation. We discuss the complete details of model training and the dataset used in Section 5 and Section 3, respectively.

☑ B1. Did you cite the creators of artifacts you used?
Section 3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Conclusion (Section 7)

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. We are using publically available datasets from mlqe-pe github repository licensed under the CC-0

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3 (Datasets)

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
All libraries used are referred to or discussed in the paper. (Section 5)

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.